



Breadth First Search: Shortest Reach

Link: <https://www.hackerrank.com/challenges/bfsshortreach>

Solution:

C++: <http://ideone.com/yccE11>

Python: <https://ideone.com/4dnOGV>

Java: <https://ideone.com/ZIHAvb>

Tóm tắt đề: Cho một đồ thị vô hướng có n đỉnh. Đỉnh được đánh số từ 1 đến n . Khoảng cách giữa 2 đỉnh luôn có độ dài là 6.

Bạn có q câu truy vấn, mỗi câu truy vấn bạn tìm chi phí đường đi ngắn nhất từ đỉnh s cho trước đến tất cả các đỉnh còn lại. In ra -1 nếu không có đường đi giữa 2 đỉnh.

Input

Dòng đầu tiên in số lượng câu truy vấn q ($1 \leq q \leq 10$), mỗi câu truy vấn gồm có:

- Dòng 1 chứa 2 số n ($2 \leq n \leq 1000$) và m ($1 \leq q \leq n*(n-1)/2$) với (n là số nút còn m là số cạnh).
- m dòng tiếp theo là danh sách các cạnh mỗi dòng 2 số u, v có kết nối với nhau ($1 \leq u, v \leq n$).
- Dòng cuối trong 1 câu truy vấn chứa đỉnh s là đỉnh cần tìm đường đi ngắn nhất từ đỉnh s này đến tất cả các đỉnh còn lại ($1 \leq s \leq n$).

Output

Chứa $n-1$ số mỗi số cách nhau dấu cách, mỗi số là đại diện cho chi phí đi từ đỉnh s đến các đỉnh trong đồ thị.

2	6 6 -1
4 2	-1 6
1 2	
1 3	
1	
3 1	
2 3	
2	

Giải thích: Có 2 truy vấn:

Truy vấn đầu: có 4 đỉnh và 2 cạnh. Đỉnh 1 nối với đỉnh 2, đỉnh 1 nối với đỉnh 3. Tìm đường đi ngắn nhất từ đỉnh 1 đến các đỉnh còn lại.

Đỉnh 1 \rightarrow 2: $1 * 6 = 6$.

Đỉnh 1 \rightarrow 3: $1 * 6 = 6$.

Đỉnh 1 \rightarrow 4: -1 (Do không có đường đi).

Truy vấn sau: có 3 đỉnh và 1 cạnh. Đỉnh 2 nối với đỉnh 3. Tìm đường đi ngắn nhất từ đỉnh 2 đến tất cả các đỉnh còn lại.

Đỉnh 2 \rightarrow 1: -1 (Do không có đường đi).

Đỉnh 2 \rightarrow 3: $1 * 6 = 6$.

Hướng dẫn giải: Bài này áp dụng BFS cơ bản, đọc vào số lượng đỉnh của đồ thị và danh sách cạnh. Sau đó chạy BFS bắt đầu từ điểm s.

Viết một hàm đếm các cạnh đi qua từ s đến các đỉnh khác. Lấy kết quả đếm $* 6$ để ra kết quả cần tìm.